

HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets

Raed A. Hasan^{*1}, Mostafa A. Mohammed², Zeyad Hussein Salih³,
Mohammed Ariff Bin Ameen⁴, Nicolae Țăpuș⁵, Muamer N. Mohammed⁶

^{1,4,6}Faculty of Computer System and software Engineering, University Malaysia Pahang,
Kuantan-26300, Pahang, Malaysia, Tel: +609-4245000, Fax: +609-4245055

^{2,5}Faculty of Automatic Control and Computers, University Politehnica of Bucharest,
313 Splaiul Independenței, 060042, Romania, Tel: +40 21 402 9100

³Faculty of petroleum and minerals engineering| Tikrit University, Tikrit-34001, Iraq

*Corresponding author, e-mail: raed.isc.sa@gmail.com

Abstract

Mobile Cloud Computing (MCC) is an emerging technology for the improvement of mobile service quality. MCC resources are dynamically allocated to the users who pay for the resources based on their needs. The drawback of this process is that it is prone to failure and demands a high energy input. Resource providers mainly focus on resource performance and utilization with more consideration on the constraints of service level agreement (SLA). Resource performance can be achieved through virtualization techniques which facilitates the sharing of resource providers' information between different virtual machines. To address these issues, this study sets forth a novel algorithm (HSO) that optimized energy efficiency resource management in the cloud; the process of the proposed method involves the use of the developed cost and runtime-effective model to create a minimum energy configuration of the cloud compute nodes while guaranteeing the maintenance of all minimum performances. The cost functions will cover energy, performance and reliability concerns. With the proposed model, the performance of the Hybrid swarm algorithm was significantly increased, as observed by optimizing the number of tasks through simulation, (power consumption was reduced by 42%). The simulation studies also showed a reduction in the number of required calculations by about 20% by the inclusion of the presented algorithms compared to the traditional static approach. There was also a decrease in the node loss which allowed the optimization algorithm to achieve a minimal overhead on cloud compute resources while still saving energy significantly. Conclusively, an energy-aware optimization model which describes the required system constraints was presented in this study, and a further proposal for techniques to determine the best overall solution was also made.

Keywords: mobile cloud computing, energy optimization, resource management, reliability

Copyright © 2018 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Cloud computing has made significant progress toward fulfilling the promise of computer accessibility for everyone. In this study, an energy optimization model for the computing nodes which incorporates the cloud computing systems is proposed. The work is based on the premise that the computing nodes should be active and consume power only when there is a work to be processed, otherwise should remain inactive [1]. Different approaches such as a neural network, fuzzy logic inference engine, and linear filters will be used to determine future workloads while the workloads and the simulated potential energy savings will be predicted using single and multi-tenant models of hosting cloud systems. With these, this study will present a minimized cost optimization model which will predict incoming workloads, determine the optimal system configuration, and make the necessary changes [2].

The optimizer, with the aid of the developed cost model and runtime constraints, will generate the minimum energy configuration of cloud computing nodes while ensuring that the performance is guaranteed. The three key areas of energy, performance, and reliability will be covered by the cost functions [3-5]. A decrease in the number of active servers was evidenced through simulation to have reduced power consumption by 42 %. Also, the simulation study showed that the inclusion of the suggested algorithms reduced the required number of calculations by over 20% when compared to the conventional static method. This ensured the

attainment of a minimal overhead on the cloud resources by the optimization algorithm, while saving energy substantially. An overall energy-aware optimization model which described the required system constraints and proposed the techniques for the determination of the best overall solution was proposed.

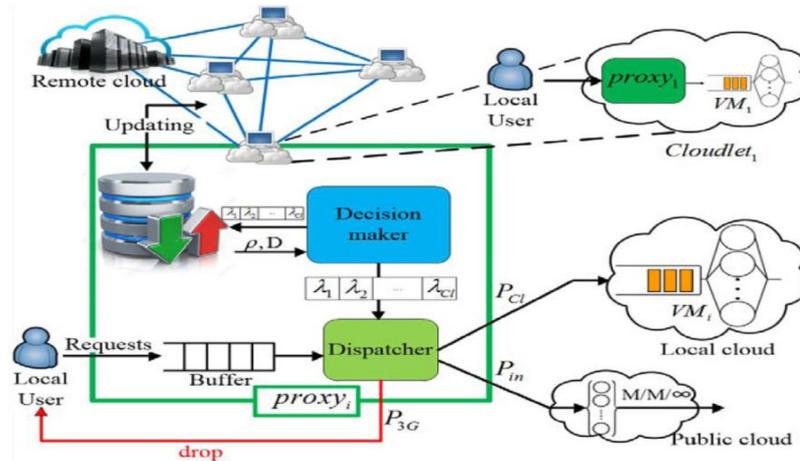


Figure 1. Shows level as Cloudlet, Local cloud and Public cloud (S. Rashidi, S. Sharifian.2017)

The main challenges of MCC are as following:

- Automated service provisioning
- Virtual Machines Migration
- Energy Management
- Stored data management
- Emergence of small data centres for cloud computing

The main reason behind focusing on energy efficiency is due to the increase in the energy cost spent on data centers. One of the major contributor to the increased energy cost is the server machine. It contributed significantly to increased energy utilization due to the direct power consumption of the cooling gadgets used in it. About 75% of the energy cost in the data center is accounted for by the direct and indirect power consumed by the server machine and cooling equipment, respectively. Furthermore, the high-performance multi-core processors used in the server machine contributes to the increased demand for power, as well as the dissipation of a considerable amount of heat.

The following works have already been contributed:

1. "Proposal of a new Hybrid algorithm named (HSO) using ACO and PSO search".
2. "Analysis of job creation time, task creation time, destruction time, result retrieval time and total time for HSO algorithm".
3. "Performance comparison of an HSO algorithm and CSO algorithms".
4. "Makespan improvement comparison of an HSO algorithm with CSO & ABC algorithms".
5. "Energy comparison of HSO algorithm and Cuckoo algorithm".

This research is organized into sections, section 1 is a preface where the problem and the most important goals have been defined. The second section went through the studies related to the problem and discussed their strengths and weaknesses. The third section provided a brief discussion of the suggested methodology, a diagram, and a mathematical equation, and the fourth section discussed the results and their comparison to the existing algorithms. The last part is a conclusion.

2. Related Works

Many solutions to the VM placement problem in the cloud have been presented in the research literature. Most implement some form of the utility function, which manages certain aspects of the cloud. The use of VM live migration to consolidate VMs onto PMs is a common

approach. Most of the solutions implement some form of prediction algorithm in order to have an idea of how much capacity is required in the future. Almost all approaches involve some form of AI heuristic since the problem of VM placement is recognized to be generally NP-Hard [6,7].

2.1. Swarm Intelligence SI

SI describes a new field of optimization methods inspired by the natural behavior of swarm insects such as ants, birds, bees and many more. Such a behavior is exploited with simple rules and embedded in a metaheuristic solution approach. The bottom line of a metaheuristic procedure is the iterative quality improvement of initial (sometimes random) candidate solutions. The combination of simple rules, within a metaheuristic, enables them to tackle combinatorial problems. Some of those are non-deterministic polynomial-time hard problems (NP-hard).

Reported research and practice results for those techniques encouraged researchers for a steady intensification and extension for the field of swarm intelligence and metaheuristics [7,8]. The history of metaheuristics and swarm intelligence is very much intertwined. Among the first and highly acknowledged metaheuristics are Genetic Algorithms (GA) [9,10], Simulated Annealing (SA) [11] and Tabu Search (TS) [12]. These are classified as metaheuristics, such as SI. Regarding the SI, the most important SI techniques are the Ant Colony Optimization (ACO) [13] and Particle Swarm Optimization (PSO) [14-17].

2.2. Swarm Intelligence used for VM placement.

An energy-efficiency evaluation model with two influencing factors (CPU and GPU utilization rates) has been proposed by Jiang, Feng, Zhao, & Li (2017) after studying the energy and performance constrained VM scheduling with data-intensive jobs. To optimize the VM selection and allocation policies, they developed an artificial bee colony algorithm and conducted several simulation studies to validate the analytical results [7].

Ghasemi-Falavarjani et al. (2014) suggested a two-stage approach for solving the problem: the first used a non-dominated Sorting Genetic Algorithm II (NSGA-II) to generate the Pareto solution set before employing entropy weight plus Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) technique to determine the optimal compromise solution. However, a context-aware offloading middleware was enhanced for contextual information collection and handling of the offloading procedure [18,19]. As a core cloud computing technique, the VM is regarded as an efficient method towards building a cloud-based data center to ensure green computing as suggested by Aksanli et al. (2012).

Gai et al. (2016) attempted an extension of the functions of cloudlets and later used a dynamic energy-aware cloudlet-based mobile cloud computing model (DECM) to further investigate their energy-aware performances in the dynamic networking environment [4].

Ranjbari & Torkestani (2017) suggested a power and SLA efficient algorithm for resource allocation through the optimization of the energy utilization, SLA violations, and the number of VM migrations. In their study, the environment parameters were adapted using Learning Automata to prevent a violation of the SLA and avoid physical machine overload [20].

2.3. Hybrid swarm algorithms used in a cloud environment

Mashayekhy et al. (2015) provided the model of a heterogeneous VM provisioning problem as a coalitional game and suggested a core-based pricing method which achieved an optimal solution. The proposed method ensured an optimal social welfare without considering the proofs of the strategy [8]. Mohammed & Tapus (2017) emphasized on efficient resource management with a focus on minimal energy absorption by using enthalpy-based CSO frameworks. They developed and resolved the resource management problems in this study. As such, different effective measures were formulated for each task, and regarding the computed enthalpy values, the CSO was used to optimally prioritize the resources [21].

The ABC was proposed by Goudarzi et al. (2017) as a fast hybrid multi-site computation offloading solution which establishes the offloading solution in a time-efficient manner. The provided solutions which include 2 decision algorithms for achieving optimal or near-optimal solutions for multi-site offloading with regard to the size of the application. The heterogeneity of a system has an important role in the determination of the dynamics of strategy-proof mechanisms [5]. Our paper presents a Hybrid Swarm Optimization (HSO) algorithm which

initiates with a PSO, and if the PSO failed to find the global optima, then, the ACO will strive to address the drawbacks of PSO.

3. Proposed Methodology

3.1. Resource allocation

This resource allocation is the procedure toward doling out and scheduling accessible resources in a most effective and economical manner. The suggested optimized HSO, accompanied by ACO and PSO solves the resources allocation problems and results in an effective resource management. In the mobile cloud computing, resource management is imperative in computing resource distribution between the customers. An efficient resource management in a data-center has its part for both consumer satisfaction and maximizing profit.

The suggested HSO, along with the energy-aware model, aids cloud administrators in the direction of choosing the priority among the clients and dispense resources effectively. This suggested resource allocation procedure is more efficient to that of the existing methods because, in those frameworks, there is no consideration of load balancing over and above energy reduction amongst the tasks. The solutions presented will potentially allow the operator of a cloud to automatically reconfigure the computer resources of the cloud in such a way as to minimize the overall energy consumption of the resources and at the same time, maintaining the performance and reliability concerns of the cloud operator.

3.2. Low Energy Optimization

A presentation of the full cost model was made relating to the areas of energy, performance, and reliability. Each area was examined and an in-depth cost model relating to all the variables and constraints (comprising the specific areas) was presented. The cost model was followed by an explanation and mathematical summary of the required operational constraints around the cost functions. Lastly, an optimization model was presented using genetic algorithms. A process flow diagram was presented which described the overall optimization algorithm.

The algorithm comprised of the workload prediction, grouping, and sorting the VMs and PMs into the category lists of VM source candidates, VM target candidates, and PM sleep candidates. Next, the lists were used to create an initial configuration of an insect. The insect is passed into the cost model for overall cost determination. The HSO algorithm then mutates the PM and VM lists, then, the processes are repeated. When the delta minimum cost is within a target epsilon value or a maximum number of iterations is reached, the algorithm exits, and the minimum cost configuration is passed into the reconfiguration process for the modification of the runtime configuration. The HSO algorithm mutates the three lists by randomly rearranging the ordering of the systems and the process is repeated until the target cost threshold is reached. Figure 2 showed the process flow diagram of the low energy optimization model implementing the HSO heuristic described in this section.

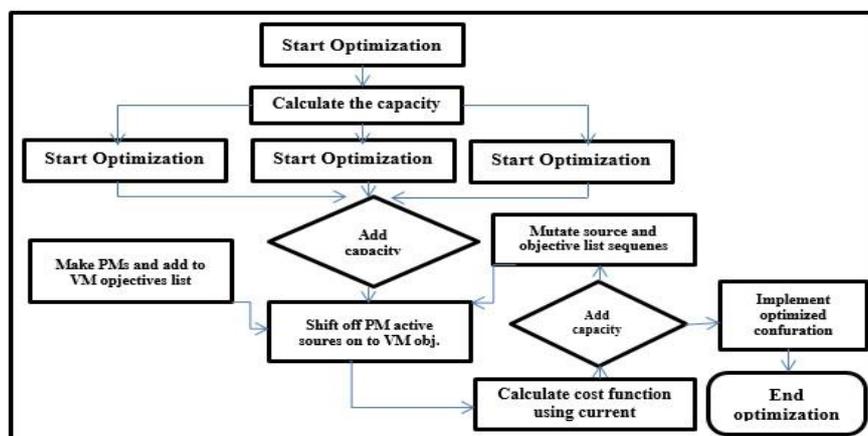


Figure 2. Low Energy Optimization Model Process Flow Diagram

3.3. The metaheuristic search (ACO & PSO) algorithms

The essential parts of an ant algorithm are the ants' solution construction and the pheromone update. Even if denoted as optional, an LS procedure is implemented in almost every metaheuristic ACO algorithm. The ACO metaheuristic algorithm is described in three basic steps. Reviewing the characteristics for ACO, it becomes obvious that artificial ants build solutions out of a finite set of possible partial solutions [5,22]. The starting point is an empty partial solution, which is extended to a feasible solution for each iteration step. The probability for choosing a solution component (or the next node) relates to the probabilistic path search behavior. This feasible solution is defined as a set of available solutions which must fulfill certain constraints.

The PSO was inspired by the natural swarm behavior of birds [21,23]. It has several advantages for solving computational tasks. When compared to the other known frameworks, the PSO is a simple concept which can be easily implemented though with a considerable level of complexity in handling several optimization tasks. Regarding competitiveness, the PSO is computationally cost-effective and fast enough. Considering the PSO algorithm, position and velocity computation resemble the basic PSO algorithm. The position calculation is smoothed by an additional velocity factor [24,25]; that is, a factor is activated in case the average swarm velocity is higher than the maximum velocity. A comparison to the basic PSO and PSO modifications proves HSO outperformed the known methods. Particularly, while grouping and exchanging prove to work against premature convergence, the relocation approach does not seem to have an impact on the solution quality. A drawback remains that they often tend to lead to suboptimal solutions; hence, a combination with an improved algorithm is considered to provide higher quality solutions [26,27].

3.4. Derivation of the Energy Cost Function

In this section, many sources of energy consumption in cloudlets were reviewed. The first is related to the number and state of all the PMs for the time duration under consideration. The PMs in the active state consume power in a linear model [18,20]. When the PMs are in the ACTIVE state but are not being utilized, the PMs can still consume 50% of peak power. Alternatively, when the PMs are placed in the SLEEP state, power consumption drops to 10-15% of peak power [27]. This illustrates the necessity of keeping as many PMs as possible in the SLEEP state in order to minimize the power consumed by the cloudlet [15]. A linearized model for power consumption was presented and used in this paper to relate PM power consumption to PM CPU utilization and is shown below.

$$P_{Active} = kU_{CPU} + P_{IDLE} \quad (1)$$

P_{Active} is related to a constant, k times the current CPU utilization plus the idle power of the system, k = Relative weighting constant for cost function. In practice, the constant k is determined for each unique class of servers through observations prior to the server class being placed in the cloudlet. This gives the total power for active PMs and the total power for the inactive as:

$$p_{total}^{Active} = \sum_i^M [k_i U_{i,0}^p (p_i) + p_i^{idle}] x_i \quad (2)$$

$$p_{total}^{Inactive} = \sum_i^M p_i^{sleep} (1 - x_i). \quad (3)$$

We can now calculate the watt-hours for running the cloudlet nodes during the current time period as:

$$\lambda_{PM}(x) = p\$ (p_{total}^{Active} + p_{total}^{Inactive}) T \quad (4)$$

where $p\$$ is the relative cost of power in \$/watt-hours delivered to the PMs for the current time period, P =set of PMs in current scope, U =utilization function, λ =cost function, k =relative weighting constant for cost function, X =configuration of PMs and VMs in the Cloudlet.

The thermal effect of the position of the active PMs in the Cloudlet is to potentially cause thermal gradients to form where PMs are clustered. When areas of the Cloudlet have

large thermal gradients, the cooling equipment tasked with dissipating the thermal energy operates less efficiently than if there are little or no gradients. Each PM in the Cloudlet monitors the temperature in its case and has the knowledge of the xyz coordinates for each PM which is kept by the cloud controller. This allows the temperature gradient to be found for each PM in the Cloudlet. This is then used to determine the cost for the gradient as shown below.

$$\lambda_{THERM}(x) = \pi_{THERM} \sum_i^M \nabla \text{Temp}(p_i) \quad (5)$$

This gives the overall cost function for energy π_{THERM} =cost factor for thermal gradients. We include a cost weighting constant κ for each cost function. We derive this constant to allow the Cloudlet operator to tune the cost functions as appropriate for the needs of each Cloudlet.

$$\lambda_{Energy}(x) = k_{PM} \lambda_{PM}(x) + k_{THERM} \lambda_{THERM}(x) \quad (6)$$

λ =Cost function, κ =relative weighting constant for cost function.

3.5. Hybrid Swarm Optimization

A hybrid algorithm was built by combining the advantages of the ACO and PSO for solving combinatorial optimization problems. Despite the avalanche of efforts already made for a competitive QAP solution method, other metaheuristics have shown superiority and robustness over the ACO [25]. Therefore, this study focused on the building of a Hybrid Swarm Optimization (HSO) framework and according to the presented metaheuristic design, the HSO can work together with other frameworks like PSO or ACO.

The extension of the Cuckoo Search Optimization (CSO) has always been conceived just to compensate its drawbacks (computation time and solution quality) with other frameworks. Recall that exploration and exploitation are two important attributes for a competitive optimization algorithm performance [13,16]. Figure 3 shows the flowchart of the HSO algorithm. The PSO takes the initial value of the solutions as size = 5050. Each insect has two properties, velocity, and position. Many iteration PSO finds the global optima while ACO enhances the cloudlet database and works to solve the drawbacks of PSO in getting the highest rate of service with an optimized energy efficiency and resource management.

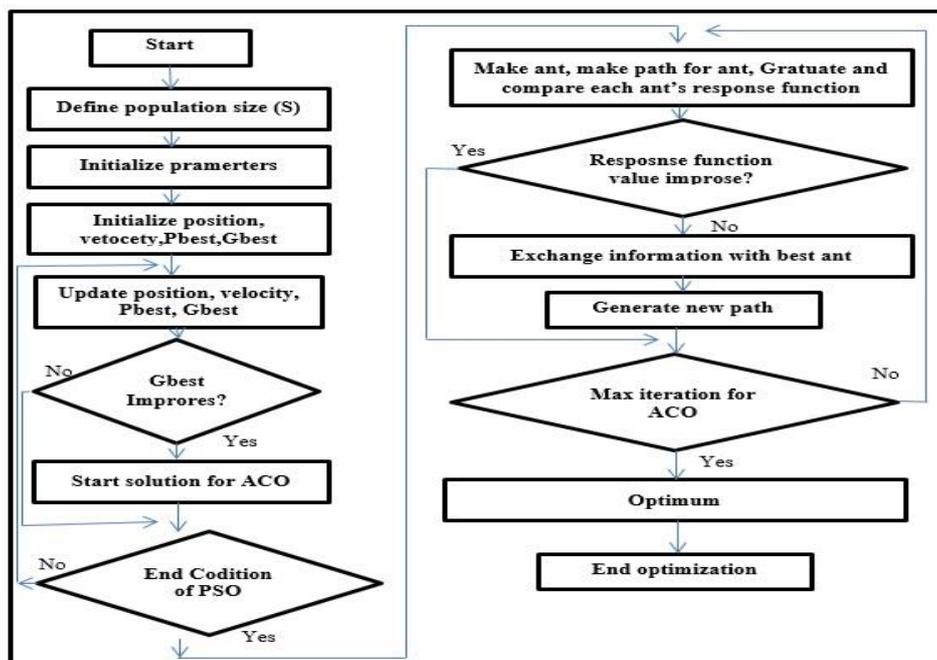


Figure 3. Flow chart of HSO algorithm

4. Results

4.1. Performance analysis

In this section, the performance of HSO was analyzed using various measures such as reduced energy consumption. The suggested HSO was used in a JAVA platform. Node loss and benchmarking with the prevailing methodologies were also evaluated. The performance of the proposed method is presented in Figures 3 to 7 and are described below.

4.1.1. Energy Consumption

The energy loss E_{loss} in a network can be determined through a multiplication of the energy loss factor with the load losses at the peak load and period T ; then, adding the obtained value to the no-load losses over the same period. It can be calculated using equation (7); The number of active servers is reduced.

$$E_{loss} = F * T * P_{loadloss\ peak} + T * P_{noloadloss} \quad (7)$$

where E_{loss} is the energy loss, F is the loss factor, T is the time period, $P_{loadloss\ peak}$ is the load losses at peak load, $P_{noloadloss}$ is the no-load losses.

The nodal energy consumption was mainly computed to avoid delays in the performance of the nodes, conserve energy, and minimize cost. At an idle nodal state, power is released and the remaining energy for the packets and the calculations denotes the power spent in the computations which took place within the routing nodes and power tunings. Equation 8 is used to determine the energy consumed by a node over time t .

$$E_{c(t)} = N_T * A + N_R * B \quad (8)$$

where $E_{c(t)}$ represents the absorbed energy by the node after time t , N_T and N_R represents the number of transmitted and received packets respectively by the node, A and B are constant factors based on the energy model. Figure 4 showed the comparison graph of the proposed HSO energy losses with that of ABO and CSO.

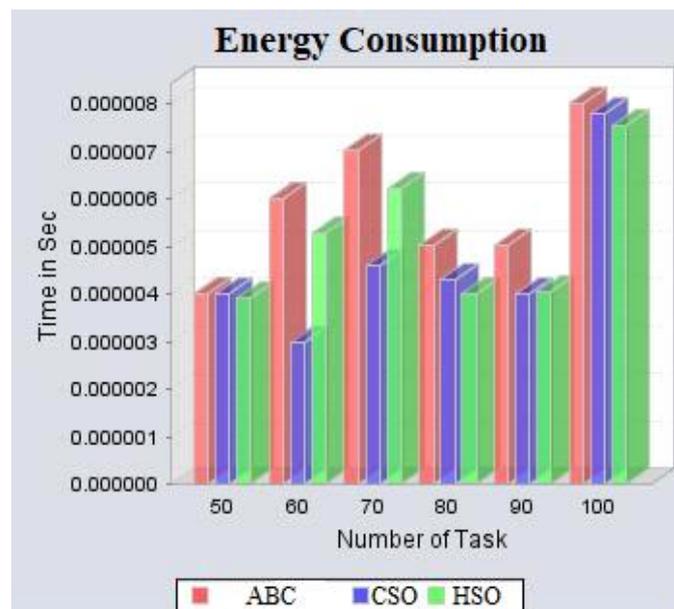


Figure 4. Comparative analysis of the suggested HSO with ABC and CSO for energy loss

4.1.2. Node loss

The only sign of a nodal failure in a network is the integration loss to the nodal processes witnessed by the other nodes [28]. Thus, nodes are considered failed when a significant component of their membership is lost. This means that while a node that caused the primary feature may not be any more, however, it is not succeeded. The node loss of the proposed framework with respect to different iteration numbers is presented in Figure 5.

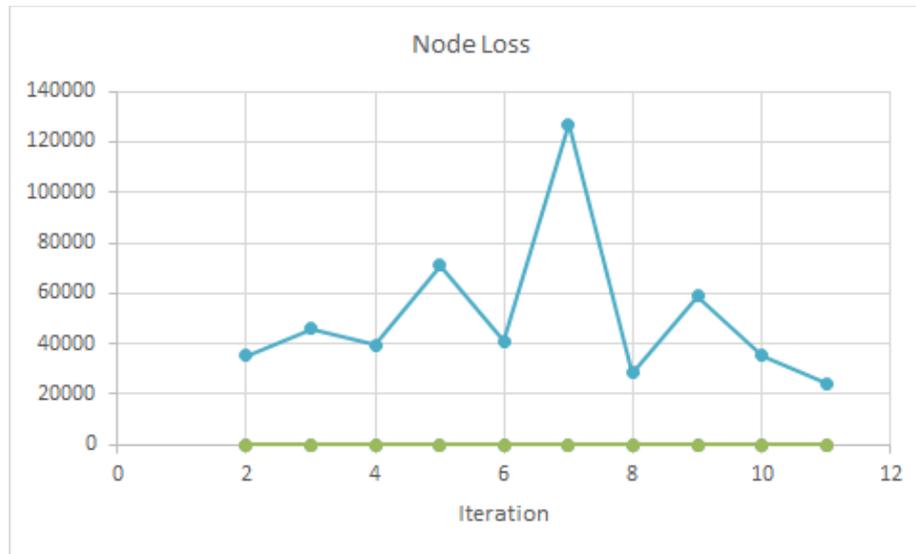


Figure 5. The graphical illustration of node loss by varying the iterations

4.1.3. Reliability Considerations for Optimal Cloud Computing

Reliability in Cloud due to repeated on-off cycles in a utility model is also considered as a benefit of the energy savings received by switching the unused PMs off. A utility function model [29] RACE has been presented which considers the MTTF, thermal effects on hardware, the cost of VM migration, and PM state. The system mitigates SLA violations based on the allocated capacity being less than the required capacity. Equation (9) is used to measure the cost of reliability. A reliability-aware server consolidation strategy focuses on characterizing and analyzing the afore-mentioned problem as a multi-objective optimization problem. Based on the utility model that was designed for the unification of the multiple constraints on the performance of SLAs, the reliability factors, and the energy cost, we have shown in Figure 5 the graph of the proposed HSO reliability.

$$\lambda_{Reliability}(x) = k_{SW}\lambda_{SW}(x) + k_{TC}\lambda_{TC}(x) \quad (9)$$

Where λ is the cost function, SW = switching, TC = thermal cycling.

4.1.4. Task Energy Consumption

Energy consumption is defined as the quantity of energy utilized by the strategy in data transfer. The task of energy consumption stands as the quantity of energy utilized to accomplish each task. The total quantity of energy utilized by a methodology in data transfer is termed the energy consumption. To do the performance breakdown of the study, the suggested HSO was compared with the existing CSO. The correlation analysis of the proposed HSO with the existing CSO is presented in Figure 6, while a comparison of the suggested HSO with CSO for energy consumption is presented in Figure 7 which demonstrates that the projected HSO achieved less energy consumption compared to the CSO as the number of task increases.

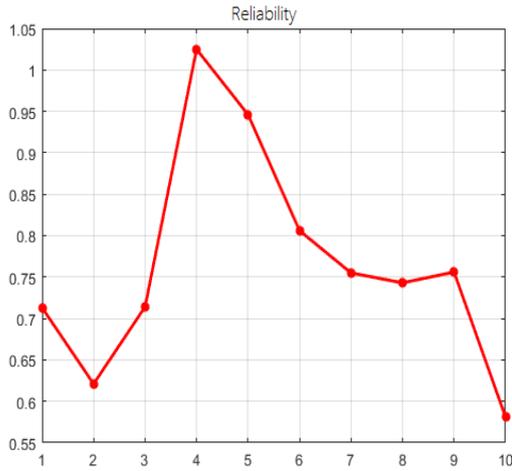


Figure 6. The graphical illustration of Reliability

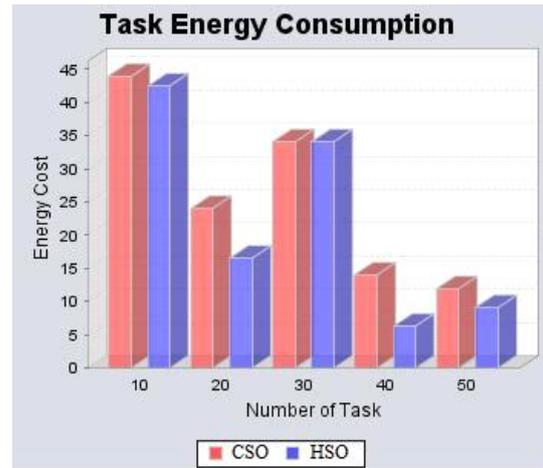


Figure 7. Comparison of the proposed HSO and CSO in terms of energy loss

4.1.5. Execution time

Execution time is the time taken from the start of the first task to the final task taken. It can be estimated using equation (10).

$$E_t = R_{first} - R_{last} \tag{10}$$

where E_t is the execution time span, R_{first} is the time of ending the last task, R_{last} is the time of starting the first task. The contrast analysis of HSO, CSO, ABC, and GA algorithms in terms of execution time is shown in Figure 8.

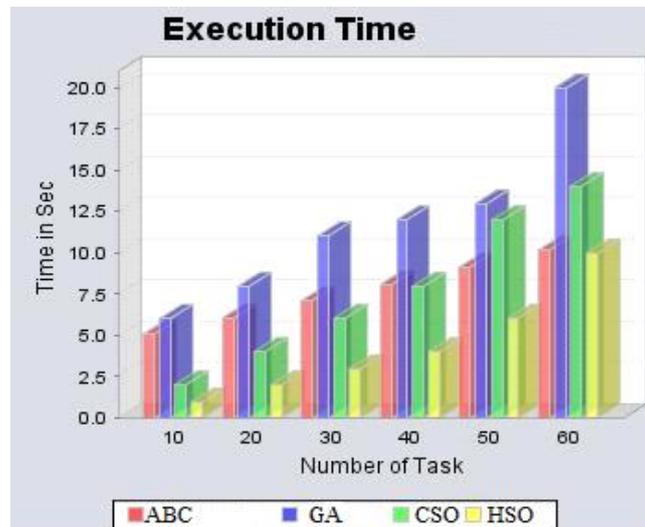


Figure 8. Comparison of HSO, ABC, GA, and CSO in terms of execution time

5. Conclusion

This paper proposed a mobility-aware resource allocation algorithm named as a hybrid swarm optimization algorithm (HSO) for the reduction of energy consumption. The work of balancing the workload among the cloudlets is another significant direction in the research for

MCC. In reality, multiple cloudlets can be set up separately in one huge venue with a high speed wired network connection among them.

The HSO algorithm optimally prioritized the resources; it requires less computing time compared to CSO algorithm. From the system model point of view, the requests of using cloud resource can be classified to the ones from both mobile users nearby and other neighbor clusters. The gains of accepting a request from a neighbor cloudlet should be balanced with the cost of increasing the systems' capacity and the cost of transferring the computing task between clusters at the same time. Lastly, the cost function of the HSO was addressed in three parts (energy, performance, and reliability) while its performance was compared to those of CSO and ABC algorithms.

6. Significance Statements

Proposal of a novel hybrid algorithm called HSO using ACO and PSO; Analysis of job and task creation time, destruction time, result retrieval time, and total time for executing the HSO. However, the limitations of wireless bandwidth and device capacity have caused various bottlenecks such as extra energy consumption and latency delay. Performance comparison of HSO and CSO algorithms; Makespan improvement comparison of HSO, CSO, and ABC algorithms; Energy comparison of HSO and CSO algorithms.

References

- [1] Beloglazov, J Abawajy, R Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*. 2012; 28: 755-768.
- [2] G Jung, MA Hiltunen, KR Joshi, RD Schlichting, C Pu. *Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures*. Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on. 2010: 62-73.
- [3] K Gai, M Qiu, H Zhao, L Tao, Z Zong. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*. 2016; 59: 46-54.
- [4] M Goudarzi, M Zamani, AT Haghghat. A genetic-based decision algorithm for multisite computation offloading in mobile cloud computing. *International Journal of Communication Systems*. 2017; 30(10).
- [5] H Grigoryan, H Astsatryan, T Gevorgyan, V Manukyan. Cloud Service for Numerical Calculations and Visualizations of Photonic Dissipative Systems. *Cybernetics and Information Technologies*. 2017; 17(5): 89-100.
- [6] RA Hasan, MN Mohammed. A Krill Herd Behaviour Inspired Load Balancing of Tasks in Cloud Computing. *Studies in Informatics and Control*. 2017; 26(4): 413-424.
- [7] J Jiang, Y Feng, J Zhao, K Li. Dataabc: a fast abc based energy-efficient live vm consolidation policy with data-intensive energy evaluation model. *Future generation computer systems*. 2017; 74: 132-141.
- [8] L Mashayekhy, MM Nejad, D Grosu. A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources. *IEEE Transactions on Parallel and Distributed Systems*. 2015; 26(9): 2386-2399.
- [9] RA Hasan, MA Mohammed, N Țăpuș, OA Hammood. *A comprehensive study: Ant Colony Optimization (ACO) for facility layout problem*. Paper presented at the Networking in Education and Research (RoEduNet). 16th RoEduNetConference. 2017
- [10] T Rashid, et al. Pm-EEMRP: Postural Movement Based Energy Efficient Multi-hop Routing Protocol for Intra Wireless Body Sensor Network (Intra-WBSN). *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2018; 16(1): 166-173.
- [11] S Kirkpatrick, CD Gelatt, MP Vecchi. Optimization by simulated annealing. *Science*. 1983; 220(4598): 671-680.
- [12] F Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*. 1986; 13(5): 533-549.
- [13] S Parsons. Ant Colony Optimization by Marco Dorigo and Thomas Stützle. MIT Press. *The Knowledge Engineering Review*. 2005; 20(1): 92.
- [14] J Kennedy. Particle swarm optimization. Encyclopedia of machine learning. Springer. 2011: 760-766.
- [15] TP Shabeera, SDM Kumar, SM Salam, KM Krishnan. Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm. *Engineering Science and Technology, an International Journal*. 2017; 20: 616-628.
- [16] RD Wang, XS Sun, X Yang, H Hu. Cloud Computing and Extreme Learning Machine for a Distributed Energy Consumption Forecasting in Equipment-Manufacturing Enterprises. *Cybernetics and Information Technologies*. 2016; 16(6): 83-97.

- [17] G Hayardisi, KB Seminar, A Ramadhan. Analysing Signal Strength and Connection Speed in Cloud Networks for Enterprise Business Intelligence. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2018; 16(4).
- [18] SG Falavarjani, M Nematbakhsh, BS Ghahfarokhi. Context-aware multi-objective resource allocation in mobile cloud. *Computers & Electrical Engineering*. 2015; 44: 218-240.
- [19] F Bacharuddin, et al. Optimum Work Frequency for Marine Monitoring Based On Genetic Algorithm. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2018; 16(4).
- [20] M Ranjbari, JA Torkestani. A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing*. 2018; 113: 55-62.
- [21] MA Mohammed, N Tapus. A Novel Approach of Reducing Energy Consumption by Utilizing Enthalpy in Mobile Cloud Computing. *Studies in Informatics and Control*. 2017; 26(4): 425-434.
- [22] JH Holland. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press. 1975: 439-444.
- [23] MA Mohammed, RA Hasan. *Particle swarm optimization for facility layout problems FLP—A comprehensive study*. Paper presented at the Intelligent Computer Communication and Processing (ICCP), 13th IEEE International Conference on. 2017.
- [24] RA Hasan, et al. Dynamic Load Balancing Model Based on Server Status (DLBS) for Green Computing. *Advanced Science Letters*. 2018; 24(10): 7777-7782.
- [25] N Moganarangan, RG Babukarthik, S Bhuvanewari, MSS Basha, P Dhavachelvan. A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach. *Journal of King Saud University - Computer and Information Sciences*. 2016; 28: 55-67.
- [26] SK Abd, SAR Al-Haddad, F Hashim, ABHJ Abdullah, S Yussof. An effective approach for managing power consumption in cloud computing infrastructure. *Journal of Computational Science*. 2017; 21: 349-360.
- [27] GAF Alfarisy, WF Mahmudy, MH Natsir. Optimizing Laying Hen Diet using Multi-Swarm Particle Swarm Optimization. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2018; 16(4).
- [28] Y Chen, L Li, H Peng, J Xiao, Y Yang, Y Shi. Particle swarm optimizer with two differential mutation. *Applied Soft Computing*. 2017; 61: 314-330.
- [29] OA Hammood, MNM Kahar, MN Mohammed. *Enhancement the video quality forwarding Using Receiver-Based Approach (URBA) in Vehicular Ad-Hoc Network*. Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), 2017 International Conference on. 2017.